L Number	Hits	Search Text	DB	Time stamp
1	8	pipeline and (stall with (stop$4 near2 clock))	USPAT; US-PGPUB	2004/01/21 14:00
-	3	(return near3 instruction) and ((multiple dual two plurality) near3 pipelines) and (stall with (return near3 address))	USPAT; US-PGPUB	2003/10/14 09:20
-	11	("3938103" | "5319757" | "5381533" | "5526498" | "5574871" | "5577217" | "5584001" | "5604877" | "5765007" | "5768576" | "5850543").PN.	USPAT	2003/06/12 10:00
-	5	douglas-jon$.in.	USPAT; US-PGPUB	2003/06/11 16:22
-	42	(return near3 instruction) and ((multiple dual two plurality) near3 pipelines) and (stall with return)	USPAT; US-PGPUB	2003/06/12 10:52
-	5	((call and return) near3 instruction) and (stall with (return near3 address))	USPAT; US-PGPUB	2003/06/11 16:34
-	7	(stall with (return near3 address))	USPAT; US-PGPUB	2003/06/11 16:35
-	6	(stack with return with address) and (stall with stack with address)	USPAT; US-PGPUB	2003/06/11 16:38
-	23	(stack with return with address) and (stall with return)	USPAT; US-PGPUB	2003/06/12 08:45
-	33	(stack with return) and (stall with return)	USPAT; US-PGPUB	2003/06/11 16:54
-	14	(stack with return) and (stall with return) not ESP	USPAT; US-PGPUB	2003/06/11 16:42
-	59	(stack with return) and ((stall bubble) with return)	USPAT; US-PGPUB	2003/06/12 08:09
-	26	(stack with return) and (bubble with return)	USPAT; US-PGPUB	2003/06/12 08:09
-	83	(stall$3 with return with (instruction address stack))	USPAT; US-PGPUB	2003/06/12 08:27
-	53	(stall$3 with return with (instruction address stack)) and call	USPAT; US-PGPUB	2003/06/12 08:24
-	109	(stall$3 same return) and (return with stack with address)	USPAT; US-PGPUB	2003/06/12 08:27
-	128	(stall$3 same return) and (return with (stack buffer) with address)	USPAT; US-PGPUB	2003/06/12 08:44
-	51	(stall$3 near5 pipeline) with return and g06f$.ipc.	USPAT; US-PGPUB	2003/06/12 08:46
-	17	(stall$3 bubble) with (return near2 address)	USPAT; US-PGPUB	2003/06/12 09:06
-	42	(return with (stack buffer cache)) same ("read-ahead" (read near2 ahead))	USPAT; US-PGPUB	2003/06/12 09:07
-	54	(return with address with predict$3) same stall	USPAT; US-PGPUB	2003/06/12 09:48
-	0	(return with address) and (return with stall)	EPO; JPO	2003/06/12 09:49
-	34	return with stall	EPO; JPO	2003/06/12 09:49
-	3	return with stall and g06f$.ipc.	EPO; JPO	2003/06/12 09:50
-	219	return with address	IBM_TDB	2003/06/12 09:51
-	0	(return with address) and (return with stall)	IBM_TDB	2003/06/12 09:51
-	0	((return with address) same stall)	IBM_TDB	2003/06/12 09:51
-	1	return with stall	IBM_TDB	2003/06/12 09:52
-	8	return same stall	IBM_TDB	2003/06/12 09:52
-	0	(return same stall) and (return with address)	DERWENT	2003/06/12 09:53
-	63	return same stall	DERWENT	2003/06/12 09:53
-	3	return same stall and pipeline	DERWENT	2003/06/12 09:54
-	43	stall with return with instruction	USPAT; US-PGPUB	2003/06/12 09:56

-		1	6151671.URPN.	USPAT	2003/06/12 09:59
-		11	(US-3938103-$ or US-5319757-$ or US-5850543-$ or US-5765007-$ or US-5768576-$ or US-5577217-$ or US-5604877-$ or US-5526498-$ or US-5381533-$ or US-5574871-$ or US-5584001-$).did.	USPAT	2003/06/12 10:00
-		4	((US-3938103-$ or US-5319757-$ or US-5850543-$ or US-5765007-$ or US-5768576-$ or US-5577217-$ or US-5604877-$ or US-5526498-$ or US-5381533-$ or US-5574871-$ or US-5584001-$).did.) and stall	USPAT	2003/06/12 10:00
-		77	712/242.ccls.	USPAT; US-PGPUB	2003/06/12 10:02
-		71	712/243.ccls.	USPAT; US-PGPUB	2003/06/12 10:02
-		110	(712/242.ccls. 712/243.ccls.) and return	USPAT; US-PGPUB	2003/06/12 10:03
-		84	(712/242.ccls. 712/243.ccls.) and (return with address)	USPAT; US-PGPUB	2003/06/12 10:03
-		17	(712/242.ccls. 712/243.ccls.) and (return with address) and (stall bubble)	USPAT; US-PGPUB	2003/06/12 10:03
-		0	((multiple dual two plurality) near3 pipeline) and ((share$1 global single) near3 (return with address with (stack cache buffer)))	USPAT; US-PGPUB	2003/06/12 10:55
-		92	((multiple dual two plurality) near3 pipeline) and ((share$1 global single one) with (return with address with (stack cache buffer)))	USPAT; US-PGPUB	2003/06/12 10:57
-		5	("4394729" | "5179673" | "5276882" | "5313634" | "5355459").PN.	USPAT	2003/06/13 08:10
-		24	5604877.URPN.	USPAT	2003/06/13 08:11
-		3	clock with throttl$3 with pipeline with (stall bubble)	USPAT; US-PGPUB	2003/06/13 11:08
-		9	clock same throttl$3 same pipeline same (stall bubble)	USPAT; US-PGPUB	2003/06/13 11:35
-		17	pipeline and (clock same throttl$3 same (stall bubble))	USPAT; US-PGPUB	2003/06/13 13:14
-		0	(plurality near3 pipeline) and ((common share) with return with (stack buffer) with address)	USPAT; US-PGPUB	2003/06/13 13:17
-		9	((plurality multiple dual) near3 pipeline) and ((common share) with return with (stack buffer))	USPAT; US-PGPUB	2003/06/13 13:18
-		9	(((plurality multiple dual) near3 pipeline) superscalar) and ((common share) with return with (stack buffer))	USPAT; US-PGPUB	2003/06/13 13:21
-		187	(((plurality multiple dual) near3 pipeline) superscalar) and ((common share single one) with return with (stack buffer))	USPAT; US-PGPUB	2003/06/13 13:23
-		0	(((plurality multiple dual) near3 pipeline) superscalar) and ((common share single one) near3 (return with (stack buffer)))	USPAT; US-PGPUB	2003/06/13 13:25
-		0	superscalar and ((return with address with (stack buffer)) same "Fig. 1")	USPAT; US-PGPUB	2003/06/13 13:27
-		290	superscalar and return with address with (stack buffer)	USPAT; US-PGPUB	2003/06/13 13:34
-		0	superscalar and ((return with address with (stack buffer)) same ("fig. 1" "fig.1"))	USPAT; US-PGPUB	2003/06/13 13:34
-		43	superscalar and ((return with address with (stack buffer)) same access)	USPAT; US-PGPUB	2003/06/13 13:39
-		31	(share sharing common) with return with address with (stack buffer)	USPAT; US-PGPUB	2003/06/13 13:39
-		1	5864707.pn.	USPAT; US-PGPUB	2003/10/09 15:15

-	24	("4044338" | "4453212" | "4504927" | "4807115" | "4858105" | "5136697" | "5179673" | "5222220" | "5226126" | "5226130" | "5274817" | "5313634" | "5339422" | "5355459" | "5454087" | "5526498" | "5564118" | "5574871" | "5584001" | "5604877" | "5606682" | "5623614" | "5649225" | "5655098").PN.	USPAT	2003/10/09 15:27
-	17	(stall$3 bubble) with (clock pulse) with pipeline with (latch register) with stage	USPAT; US-PGPUB	2003/10/14 09:29
-	41	stall$3 with clock with pipeline with (latch register)	USPAT; US-PGPUB	2003/10/14 09:29
-	0	"pipeline latch" with stall$3 with clock	USPAT; US-PGPUB	2003/10/14 10:08
-	3	"pipeline register" with stall$3 with clock	USPAT; US-PGPUB	2003/10/14 10:09
-	4	stall$3 with pipeline with return with address	USPAT; US-PGPUB	2003/10/14 11:27
-	22	stall$3 with return with address	USPAT; US-PGPUB	2003/10/14 13:09
-	38	stall$3 with return with memory	USPAT; US-PGPUB	2003/10/14 13:09
-	4	stall$3 with (return near3 instruction) with memory	USPAT; US-PGPUB	2003/10/14 13:41
-	182	(return near2 stack) and ((plurality multiple second) near2 pipeline)	USPAT; US-PGPUB	2003/10/14 13:42
-	85	(return adj2 stack) and ((plurality multiple second) adj pipeline)	USPAT; US-PGPUB	2003/10/14 13:46
-	17	(return adj2 stack) with superscalar	USPAT; US-PGPUB	2003/10/14 13:46
-	7	stall$3 with "call instruction"	USPAT; US-PGPUB	2004/01/20 10:27
-	17	(stall$3 delay$3) with (call jump jmp) same ((stack buffer) near3 return)	USPAT; US-PGPUB	2004/01/20 10:54
-	1	"return address stack cache".ti.	IBM_TDB	2004/01/20 10:32
-	8	("4399507" | "4954947" | "5117498" | "5222220" | "5579520" | "5598560" | "5701449" | "5867696").PN.	USPAT	2004/01/20 10:49
-	15	("3231863" | "3740728" | "3840861" | "3875391" | "3900834" | "3935563" | "3984817" | "4016543" | "4025771" | "4041462" | "4047162" | "4068304" | "4109311" | "4112489" | "4187539").PN.	USPAT	2004/01/20 10:52
-	66	4399507.URPN.	USPAT	2004/01/20 10:52
-	1	(stall$3 delay$3) with (call jump jmp) same (((stack buffer) near3 return) with access)	USPAT; US-PGPUB	2004/01/20 10:55
-	169	(stall$3 delay$3) with (call jump jmp) near1 instruction	USPAT; US-PGPUB	2004/01/20 10:56
-	1	(stall$3 delay$3) with (call jump jmp) near1 instruction with access$3	USPAT; US-PGPUB	2004/01/20 10:57
-	4	share with return near3 (stack buffer)	USPAT; US-PGPUB	2004/01/20 10:59
-	372	call near4 follow$3 near4 return	USPAT; US-PGPUB	2004/01/20 11:01
-	0	(call near4 follow$3 near4 return) with stall	USPAT; US-PGPUB	2004/01/20 11:01
-	0	(call near4 follow$3 near4 return) with stall$3	USPAT; US-PGPUB	2004/01/20 11:01
-	35	(call near4 follow$3 near4 return) with delay$3	USPAT; US-PGPUB	2004/01/20 11:02
-	18	(call near4 after near4 return) with delay$3	USPAT; US-PGPUB	2004/01/20 11:03
-	0	(call near4 after near4 return) with stall$3	USPAT; US-PGPUB	2004/01/20 11:03
-	4	(call with return) with stall$3	USPAT; US-PGPUB	2004/01/20 11:07
-	50	(call with call) with stall$3 and g06f$.ipc.	USPAT; US-PGPUB	2004/01/20 11:29

-	0	stall$3 with push with (stack buffer) with return	USPAT; US-PGPUB	2004/01/20 11:29
-	5	stall$3 with push with stack	USPAT; US-PGPUB	2004/01/20 11:34
-	12	delay$3 with push with stack and g06f$.ipc.	USPAT; US-PGPUB	2004/01/20 11:35
-	11	(US-6151671-$ or US-5604877-$ or US-6247134-$ or US-5835753-$ or US-5968169-$ or US-5764946-$ or US-5222220-$ or US-5896528-$ or US-4399507-$ or US-6170998-$).did. or (NN9204269).tban.	USPAT; IBM_TDB	2004/01/21 08:46
-	0	(conflict$3 with (call jump) with instruction) same stall	USPAT; US-PGPUB	2004/01/21 10:54
-	10	(conflict$3 with (call jump) near2 instruction)	USPAT; US-PGPUB	2004/01/21 10:55

TDB-ACC-NO: NN9204269

DISCLOSURE TITLE: Return Address Stack Cache.

PUBLICATION-DATA: IBM Technical Disclosure Bulletin, April 1992, US

VOLUME NUMBER: 34

ISSUE NUMBER: 11

PAGE NUMBER: 269 - 271

PUBLICATION-DATE: April 1, 1992 (19920401)

CROSS REFERENCE: 0018-8689-34-11-269

DISCLOSURE TEXT:

- The return instruction for the Intel 80386 is basically an unconditional branch where the target address is out somewhere in memory. You must therefore first bring in the address before you can kick off the instruction fetcher. The performance loss here is in waiting for the target address. Disclosed is a design for addressing the performance loss.
- The current method used by the 80386 to execute near call returns is to issue a pop instruction to bring in the return address from the system stack and then start fetching instructions at the return address. The proposed method calls for an internal return stack to hold the return address. This allows for the fetching at the return address to occur sooner, eliminating the need for having to wait for the pop to the system stack to complete.
- The figure shows the data flow associated with the return prefetcher. What the data flow is trying to accomplish

is to mirror the system stack. Up to three call returns are saved for prefetching. The address saved is the real address of the return point (address of the instruction following the call instruction).

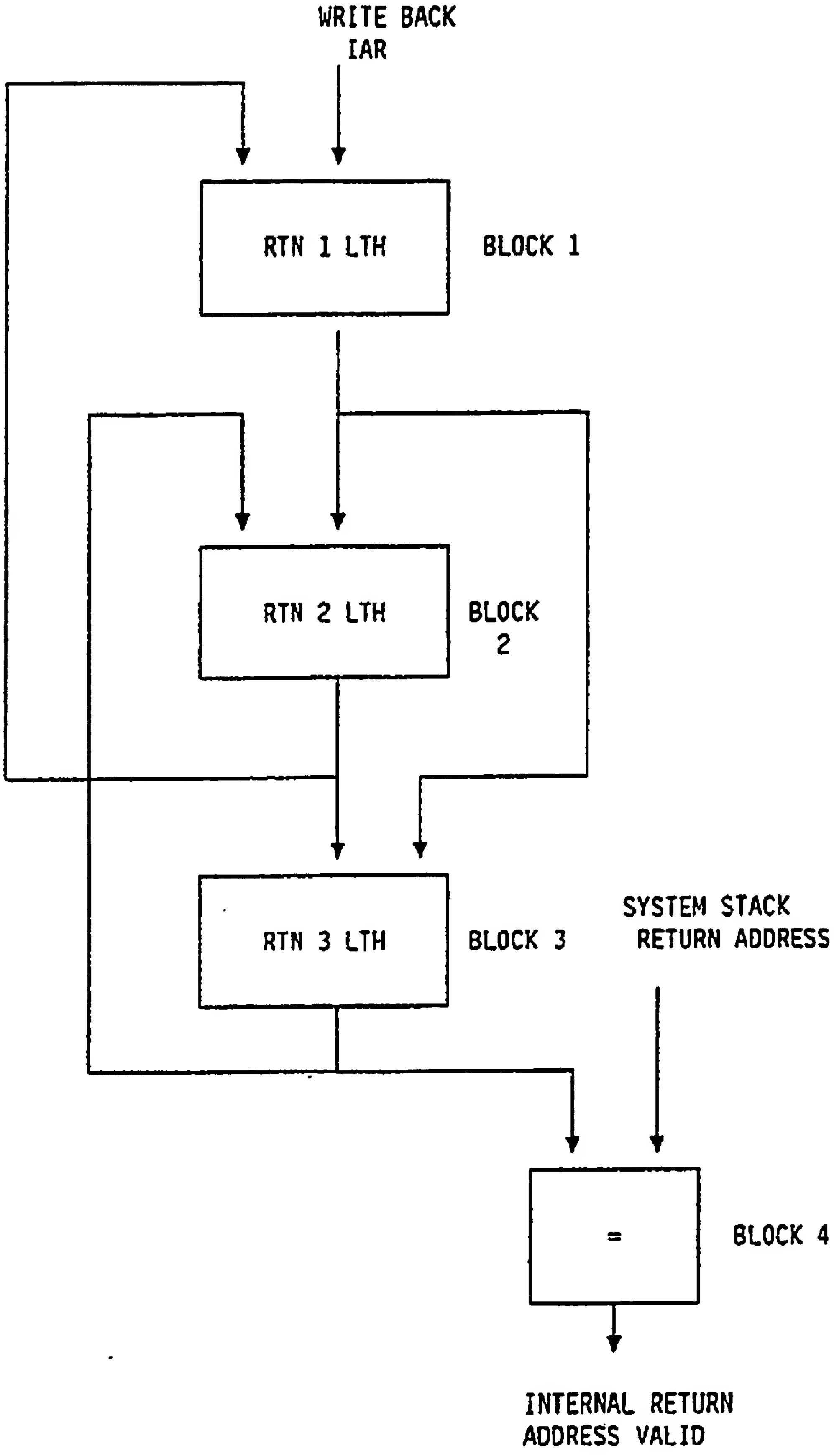
Below is the sequence of events which occur during calls.

- 1. Call instruction detected in decode stage.
- 2. Call enters write back stage, instruction pointer of call is stored (pushed) in RTN 1 or block 1.
- 3. The return instruction is detected in decode stage. The Fetcher prefetches the return address saved on the return stack. During write back (of the pop instruction) the stack is popped. The prefetched real address is also saved in RTN 3 or block 3 for validation against the real address (in block 4) in the system stack, the true return address.
- 4. Return instruction enters the execute stage. When the pop from the system stack is completed, the return address is translated (from virtual to real).
- 5. Return now enters the write back stage. The real return address from the system stack is compared against the return address from the internal return stack. This function is done to make sure no code has modified the return address in the system stack. The compare is done in block 4 and the return address prefetched is in RTN 3, block 3. If they are equal, then the prefetch was correct. If they are not equal, then purge the pipeline and start fetching with the address from step 4.
- If a call is followed by another call instruction, the return stack pushes the return address of the first call into RTN 1 (block

1). The second call also pushes its return address into RTN 1.

This, in turn, causes the return of the first call to be pushed into RTN 2 (block2). The first return pops RTN 1 (second call return address), RTN 2 goes to RTN 1. The second return pops RTN 1 (first call return address). Both the push and pop operations are done during the write back stage.

- When a call is followed by a return instruction before the call instruction enters write back stage, the prefetcher and decoder halt until the call completes write back. RTN 1 is used as the prefetch address in the cycle after write back.
- In the case of a return followed by a call instruction before the return is executed, if the prefetched return address is wrong (result of compare in block 4 is not equal), then the call needs to be invalidated in addition to purging the pipeline.
- If a return is followed by another return instruction before the first one completes, you need to hold the second return in decode until the first one completes in write back stage.
- If you detect a 4th call instruction without a return (past 3 levels of nesting), then on the 4th call, you go ahead and prefetch the call target and during write back, push the return address. You have now lost the return address of the 1st call, so what you are doing is just keeping around the last 3 returns in the stack. Now, when you detect the return instruction of the 1st call, you go ahead and use the return address from the system stack.
- Anytime the code segment (CS) is changed, by a task switch, change CS instruction, far call, etc. You must invalidate the return stack.



Return Stack Data Flow